REMARKS

Claims 1-12, 14-21, and 23-34 are pending in the present application. By this Response, claims 13 and 22 are canceled, claims 12, 14-16, 19-21, 23-25 and 28 are amended, and claims 29-34 are added. Reconsideration of the claims in view of the above amendments and the following remarks is respectfully requested.

Amendments are made to the specification to update the cross-reference information for the co-pending and commonly assigned U.S. Patent Applications. No new matter is added by the amendments to the specification.

I. 35 U.S.C. § 102, Alleged Anticipation of Claims 1-28 Based on Berry

The Office Action rejects claims 1-28 under 35 U.S.C. § 102(e) as being anticipated by Berry et al. (US Patent No. 6,539,339 B1). This rejection is respectfully traversed.

A. Claims 1-11

As per independent claim 1, the Office Action states:

> Regarding claim1 Berry anticipates, monitoring performance of a program being executed using per thread metric variables with reused kernel threads comprising;
> receiving an event indication (3:35-40);
> ascertaining kernel thread level profile information (3:40-47);
> identifying a kernel thread, wherein the kernel thread level profile information is attributed to the identified kernel thread (3:40-47);;
> determining whether the identified kernel thread has been reused (17:3-5, see recursion depth for reuse); and
> updating profile information with the kernel thread level profile information based whether the identified kernel thread has been reused (fig. 22, 2216).

Claim 1 reads as follows:

> 1. A method for monitoring performance of a program being executed using per thread metric variables with reused kernel threads comprising:
>
> receiving an event indication;
>
> ascertaining kernel thread level profile information;
>
> identifying a kernel thread, wherein the kernel thread level profile information is attributed to the identified kernel thread;
>
> determining whether the identified kernel thread has been reused; and
>
> updating profile information with the kernel thread level profile information based whether the identified kernel thread has been reused. (emphasis added)

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). Applicants respectfully submit that Berry does not identically show each and every feature of the present claims arranged as they are in the claims. Specifically, Berry does not teach determining whether the identified kernel thread has been reused or updating profile information with the kernel thread level profile information based on whether the identified kernel thread has been reused, as recited in claim 1.

Berry is directed to a method and system for maintaining a thread-relative metric for trace data using device driver support. With the method and system of Berry, a profiling process may detect a current event and, in response to the event, may request an elapsed metric since a preceding event. The profiling process then receives a thread-relative elapsed metric and may output a trace record for the current event in which is store a metric equal to the received thread-relative elapsed metric. In response to a notification of an occurrence of the event, a device driver computes the thread relative elapsed metric by determining a current thread, retrieving a stored metric for the preceding event of the current thread, obtaining a current metric, and computing the thread-relative elapsed metric as a difference between the current metric and the stored metric.

Berry does not teach determining whether the identified kernel thread has been reused, as recited in claim 1. It is a basic tenet that the specification may be used as a dictionary for the terms used in the claims and that the claims must be read in light of the specification. As noted in the present specification, the term "reused" refers to a kernel thread being referenced by at least two different application threads (see page 53, line 26 to page 54, line 11 and page 61, line 29 to page 62, line 1, for example). Thus, the present invention as recited in claim 1, when read in light of the present specification, includes the step of determining whether a kernel thread has been previously referenced by more than one application thread. The Berry reference does not teach such a feature because Berry is not directed to solving the problem associated with the accumulation of metric values for reused kernel threads. To the contrary, the Berry reference is directed to solving the problem of a thread switch occurring during an attempt to retrieve the time of an event (see column 3, lines 23-32).

The Office Action alleges that Berry teaches determining whether an identified kernel thread has been reused at column 17, lines 3-5 since this section of Berry allegedly teaches a recursion depth value and the Office Action equates this recursion depth value as being reuse of a kernel thread. Applicants respectfully disagree.

Column 17, lines 3-5 read as follows:

> The cumulative time in the example shown in FIG. 11B is four units of time. Finally, the recursion depth of call stack CAB is one, as none of the three routines present in the call stack have been recursively entered.

This section of Berry has nothing to do with whether a kernel thread is reused by application threads. To the contrary, all that this section of Berry is teaching is that the call stack may have an associated value that identifies how many times a routine is recursively entered. Recursion is the ability of a routine to call itself. Thus, the last value for each node in the tree data structure shown in Figure 11B of Berry represents how many times a routine calls itself. There is no correlation between this teaching and determining whether a kernel thread is reused by application threads. The number of times a routine is recursively entered in itself has no bearing on whether a kernel thread is reused or not. Berry simply does not teach determining whether a kernel thread is reused or not. Since Berry does not teach determining whether a kernel thread is reused, Berry cannot teach updating profile information with the kernel thread level profile information based on whether the identified kernel thread has been reused.

Thus, in view of the above, Applicants respectfully submit that Berry does not teach each and every feature of independent claim 1 as is required under 35 U.S.C. § 102(e). At least by virtue of their dependency on claim 1, Berry does not teach each and every feature of dependent claims 2-11. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 1-11 under 35 U.S.C. § 102(e).

Furthermore, Berry does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. Berry is actually directed to solving a completely different problem from that solved by the present invention and does not even mention kernel threads being reused or a mechanism for determining whether a kernel thread is being reused. Absent the Examiner pointing out some teaching or incentive to implement Berry to determine whether a kernel thread has been reused and then updating profile information based on whether the identified kernel has been reused, one of ordinary skill in the art would not be led to modify Berry to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify Berry in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the Applicants' disclosure as a template to make the necessary changes to reach the claimed invention.

In addition to their dependency on claim 1, Berry does not teach each and every specific feature recited in dependent claims 2-11. For example, with regard to claim 2, Berry does not teach checking a hash table for an entry of the identified kernel thread. The Office Action alleges that this feature is taught at column 10, lines 55-57 which read as follows:

> In post-processing phase 504 B-trees and/or hash tables may be employed to maintain names associated with records in the trace file to be processed.

While this section of Berry mentions that a hash table may be used to maintain names associated with records of a trace file, there is no teaching in this section of checking a hash table for an entry of an identified kernel thread. The general teaching of using hash tables to store names associated with records in a trace file does not teach or even suggest to one of ordinary skill in the art to check a hash table for an entry corresponding to an identified kernel thread.

Regarding claim 3, Berry does not teach applying kernel thread level profile information for a reused identified kernel thread to one of a previous application thread and a new application thread. As previously discussed above, Berry does not teach determining whether a kernel thread is reused and thus, cannot teach the features of claim 3. Moreover, the Office Action alleges that the features of claim 3 are taught by Berry in Figure 22 as element 2216. Element 2216 of the flowchart in Figure 22 reads "Device Driver Updates Previously Saved Time for Current Thread By storing Current Time as Previously Saved Time." This function associated with element 2216 has nothing to do with whether a kernel thread is reused or not and thus, cannot teach the features of claim 3 since these features reference "a reused identified kernel thread," let alone a "previous application thread" or "a new application thread."

With regard to claim 4, Berry does not teach applying the kernel thread level profile information for the reused identified kernel thread to one of a previous application thread and a new application thread, marking the previous application thread being transmitted or associating the reused identified kernel thread with the new application thread in the hash table. As mentioned above with regard to claim 3, Berry does not teach identifying a kernel thread as being reused and thus, cannot teach the features of applying the kernel thread level profile information for the reused identified kernel thread to one of a previous application thread and a new application thread.

In addition, Berry does not teach marking a previous application thread as being transmitted. The Office Action alleges that this feature is taught in Figure 23 as element 2302 which reads "Device Driver Notified of Thread Switch from Previous Thread to

Current Thread." This element does not teach marking a previous application thread as having been transmitted. To the contrary, all this element teaches is that the functionality of the flowchart illustrated in Figure 23 is initiated when the device driver is informed that a thread switch has occurred.

Moreover, Berry does not each associating a reused identified kernel thread with a new application thread in a hash table. The Office Action alleges that this feature is taught in element 2306 of Figure 23 which reads "Device Driver Computes Difference Between Current Time and Previous Event Time for Previous Thread." This has nothing to do with associating a reused kernel thread with a new application thread in a hash table. To the contrary, all this element teaches is the calculation of a difference in event times.

The distinctions of the specific features of claims 2-4 are exemplary of the deficiencies of the Berry reference. Dependent claims 5-11 recite additional features that are likewise, not taught by the Berry reference. Thus, all of claims 2-11 recite features that are not taught by Berry and are therefore allowable over Berry. Accordingly, Applicants respectfully request withdrawal of the rejections of claims 2-11 based on Berry.

B. Claims 12, 14-18, 21, 23-28

With regard to independent claims 12, 21 and 28, the Office Action merely states to refer to the rejection of claim 1 above. Applicants first wish to point out that claims 12, 21 and 28 are of a different scope than claim 1. For example, claim 12, which is representative of the other independent claims 21 and 28, reads as follows:

> 12. A method for monitoring performance of a program being executed using per thread metric variables with reused kernel threads comprising:
>
> receiving a value of a metric variable for a kernel thread;
>
> determining if the kernel thread has been previously used by a first application thread; and
>
> applying the value of the metric variable to a second application thread if the kernel thread has been previously used by the first application thread. (emphasis added)

Berry does not teach determining if a kernel thread has been previously used by a first application thread. In fact, Berry does not teach kernel threads being used by application threads and thus, cannot teach determining if a kernel thread has been previously used by a first application thread. To the contrary, the system of Berry is directed to identifying switches in application threads and resolving event timing based on the detection of an application thread switch. There is no teaching in Berry regarding kernel threads or determining if a kernel thread has been previously used by a first application thread. Since Berry does not teach determining if the kernel thread has been previously used by a first application thread, there cannot be any teaching in Berry regarding applying a value of a metric variable for a kernel thread to a second application thread when it is determined that the kernel thread has been previously used by a first application thread.

The only sections reference by the Office Action with regard to claims 12, 21 and 28 are the same sections discussed above with regard to claim 1 (since the Office Action fails to explicitly address the features of claims 12, 21 and 28 and instead rests on the rejection of claim 1). These sections have been shown above to only teach maintaining information regarding how many times a routine calls itself (recursive entry). Recursive entry of a routine has nothing to do with determining whether a kernel thread has been previously used by a first application thread and then applying the value of a metric variable to a second application thread if the kernel thread has been previously used by a first application thread.

In view of the above, Applicants respectfully submit that Berry does not teach each and every feature of independent claims 12, 21 and 28. At least by virtue of their dependency on claims 12 and 21, respectively, Applicants respectfully submit that Berry does not teach each and every feature of dependent claims 14-18 and 23-27. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 12, 14-18, 21 and 23-28 under 35 U.S.C. § 102(e).

Furthermore, Berry does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. Absent the Examiner pointing out some teaching or incentive to implement Berry to determine whether a kernel thread has been previously used by a first application thread and then applying the value of the metric variable to a second application thread if the kernel thread has been previously used by the first application thread, one of ordinary skill in the art would not be led to modify Berry to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify Berry in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the Applicants' disclosure as a template to make the necessary changes to reach the claimed invention.

In addition to the above, Berry does not teach the specific features recited in dependent claims 14-18 and 23-27. For example, with regard to claims 14 and 23, Berry does not teach identifying the first application thread as being terminated based on the kernel thread having been previously used by the first application thread. The Office Action alleges that the features of these claims are taught by Berry in Figure 23, elements 2302 and 2306, the text of which is reproduced above (see rejection of claim 4 discussed above). Neither of these steps in the flowchart of Figure 23 have anything to do with identifying an application thread as being terminated based on a determination that a kernel thread has been previously used by the application thread.

As a further example, with regard to claims 16 and 25, Berry does not teach comparing an identity of a kernel thread to a list of identities of previously used kernel threads. The Office Action again alleges that this feature is taught in element 2306 of Figure 3 in Berry. As discussed above, all this element teaches is that a device driver computes the difference in a current time and a previous event time for a previous thread. It is not at all clear how this has anything to do with comparing the identity of a kernel thread to a list of identities of previously used kernel threads.

The distinctions of the specific features of claims 14, 16, 23 and 25 are exemplary of the deficiencies of the Berry reference. Dependent claims 15, 17-18, 24 and 26-27 recite additional features that are likewise, not taught by the Berry reference. Thus, all of claims 14-18 and 23-27 recite features that are not taught by Berry and are therefore allowable over Berry. Accordingly, Applicants respectfully request withdrawal of the rejections of claims 14-18 and 23-27 based on Berry.

C. Claims 19-20

With regard to independent claim 19, the Office Action merely states to refer to the rejection of claim 1 above. Applicants first wish to point out that claim 19 is of a different scope than claim 1. Claim 19 reads as follows:

> 19. A method for monitoring performance of a program being executed using per thread metric variables with reused kernel threads comprising:
>
> receiving a plurality of values of metric variables for a plurality of kernel threads;
>
> for each kernel thread:
>
> determining if a kernel thread has been previously used by a prior application thread; and
>
> applying the value of the metric variable for the kernel thread to a current application thread if the kernel thread has been previously used by a prior application thread. (emphasis added)

As previously noted above, Berry does not teach determining if a kernel thread has been previously used by an application thread. Berry also does not teach applying a value of a metric variable for a kernel thread to a current application thread if the kernel thread has been previously used by a prior application thread. The Examiner is referred to the above discussed of independent claims 12, 21 and 28, which recite similar features. The teaching of maintaining information regarding recursive entries into a routine has nothing to do with the features emphasized above in claim 19, as previously discussed.

Therefore, Applicants respectfully submit that Berry does not teach each and every feature of claims 19-20 as is required under 35 U.S.C. § 102(e). Accordingly, Applicants respectfully request withdrawal of the rejection of claims 19-20 under 35 U.S.C. § 102(e).

II. Newly Added Claims 29-34

Claims 29-34 are added to recite additional features of the present invention. Specifically, claims 29, 31 and 33 are added to recite that the value of the metric variable for the kernel thread is received in response to an event occurring based on the execution of the second application thread. Claims 30, 32 and 34 are added to recite that applying the value of the metric variable to the second application thread includes increasing a current metric value for the second application thread in a node associated with the second application thread by an amount equal to the value of the metric variable. Support for the addition of these claims may be found at least in Figures 21 and 22 of the present specification. Berry does not teach any of these recited features.

III. Conclusion

It is respectfully urged that the subject application is patentable over Berry and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Respectfully submitted,

DATE: September 3, 2003

Stephen J. Walder

Stephen J. Walder, Jr.
Reg. No. 41,534
Carstens, Yee & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001
Attorney for Applicants